NASA/CR—2002–211406

Activity Tracking for Pilot Error Detection from Flight Data

Todd J. Callantine
San Jose State University, San Jose, California

June 2002

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR—2002–211406

Activity Tracking for Pilot Error Detection from Flight Data

Todd J. Callantine
San Jose State University, San Jose, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

Acknowledgments

This work was funded under the System Wide Accident Prevention element of the FAA/NASA Aviation Safety Program. Thanks to the NASA Langley B757 Flight Test team for their assistance with data collection.

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
301-621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

Introduction

Problems associated with human error have long been recognized (e.g., Babbage, 1961). More recently, Perrow (1984) characterized how high system complexity contributes to accidents and, together with the introduction of 'glass cockpit' aircraft (e.g., Wiener and Curry, 1980), invigorated interest in human error. Reason's (1990) theoretical treatment marks the beginning of human error research on several fronts. One sort is devoted to the collection and analysis of data on human error and its effects. For example, Johnson (1998) investigates methods for analyzing temporal features of incidents, as well as new ways to report incidents (Johnson, 2000). Accessing information in a large database of incident reports has in turn led to research on advanced search tools (McGreevy, 2001).

Another research area focuses on formal methods that can help reveal potential error-related problems during the design process. For example, Degani and Heymann (2000) use formal specifications of system behavior to identify unsafe interface abstractions. Sherry *et al*. (2001) use a formal system model to explain how operators misunderstand a system, and how it might be redesigned. Formal task representations also enable scrutiny of human-error tolerance (Wright, Fields, and Harrison, 1994) and temporal aspects of operator, system, and environmental behavior (Fields, Wright, and Harrison, 1996). A methodology for analyzing the potential for human error during the design process also incorporates some of these ideas (Fields, Harrison, and Wright, 1997). Johnson (2001) examines how error reporting can be used to support system refinement in the initial stages of implementation when the design is still in flux.

Models of human operators anchor two additional areas of research. One uses engineering-oriented computational models as the basis for preventing error and improving error recovery by training and later aiding the operator (e.g., Mitchell, 2000). Another research area seeks to develop models, either theoretical (e.g., Busse and Johnson, 1998) or computational (e.g., Byrne and Bovair, 1997), that can elucidate the cognitive bases of human error.

This report describes an application of the Crew Activity Tracking System (CATS) that could contribute to future efforts to reduce flight crew errors. It demonstrates how CATS tracks crew activities to detect errors, given flight data and air traffic control (ATC) clearances (already provided, in some cases, by digital data link communication technology, e.g., Smith, Brown, Polson, and Moses, 2001). CATS implements a so-called 'intent inference' technology, called *activity tracking*, in which it uses a computational 'engineering' model of the operator's task, together with a representation of the current operational context, to predict nominally preferred operator activities and interpret actual operator actions.

CATS, too, has its roots in glass cockpit aircraft automation research. It was originally implemented to track the activities of Boeing 757 pilots, with a focus on automation mode errors (Callantine and Mitchell, 1994). The CATS activity tracking methodology was validated as a source of real-time knowledge to support a pilot training/aiding system (Callantine, Mitchell, and Palmer, 1999). CATS is useful as an analysis tool for assessing how operators use procedures developed to support new operational concepts (Callantine, 2000a, 2000b). It also serves as a framework for developing agents to represent human operators in incident analyses and distributed simulations of new operational concepts (Callantine, 2001a).

The research described here draws in large part from these earlier efforts. In particular, the CATS model of B757 flight crew activities has been expanded and refined. The representation of operational context used to reference the model to predict nominally preferred activities has similarly undergone progressive refinement. And, while the idea of using CATS to detect flight crew errors from flight data is not new, this report presents an example of CATS detecting a genuine, in-flight crew error from actual aircraft flight data.

Using CATS to detect errors from flight data has several potential benefits (Callantine, 2001b). First, CATS provides information about procedural errors that do not necessarily result in deviations, and therefore would not otherwise be reported. Second, CATS enables airline safety managers to 'automatically' incorporate information about a detected error into a CATS-based training curriculum. Other pilots could 'relive' a high-fidelity version of the context in which another crew erred. Increasing the efficiency and fidelity of information transfer about errors to the pilot workforce in this way would likely yield safety benefits.

It is important to note that flight crews need not view such an application as punitive. It is incumbent on airline safety and training managers to ensure that the CATS model used to detect errors *exactly* matches the training provided to flight crews. Research indicates that much of what pilots know about some autopilot functionality currently is not formally trained (Mitchell, 2000). Thus, a safety-enhancement program that uses CATS to detect errors would improve training by requiring safety and training managers to explicate policies about how an aircraft should preferably be flown.

The report is organized as follows. It first describes the CATS activity tracking methodology, and information flow in CATS. The report then describes the CATS implementation for detecting pilot errors. It first describes flight data obtained for this demonstration from the NASA Langley Boeing 757 (B757) Airborne Research Integrated Experiment System (ARIES) aircraft. It next describes two key representations. The first is a portion of a CATS model of B757 flight operations. The second is a representation of the constraints conveyed by ATC clearances that plays a key role in representing the current operational context (Callantine, 2002). An example from the available flight data then illustrates CATS detecting pilot errors. The report concludes with a discussion of future research challenges.

Activity Tracking

Activity tracking is not merely the detection of operational 'deviations'. The activity tracking methodology involves first predicting the set of expected nominal operator activities for the current operational context, then comparing actual operator actions to these expectations to ensure operators performed correct activities. In some situations, various methods or techniques may be acceptable; therefore the methodology also includes a mechanism for determining that, although operator actions do not match expectations exactly, the actions are nonetheless correct. In this sense, CATS is designed to 'track' flight crew activities in real time and 'understand' that they are error-free. As the example below illustrates, 'errors' CATS detects include those that operators themselves detect and rapidly correct; such errors may nonetheless be useful to examine.

In addition to parameters that define the state of the controlled system, activity tracking also requires data about the dynamic set of constraints on controlled system behavior, as well as data about actual operator actions. For flight deck applications, constraint data in the form of data linked ATC clearance information will likely be widely available in the near future, as noted above, but a number of legal issues impede the release of pilot action data (U.S. GAO, 1998). This report takes the view that the promise of significant safety benefits, together with anonymity provisions similar to those of the Aviation Safety Reporting System (ASRS), can help overcome these issues in the future. Activity tracking also requires a valid model of nominally correct operator activities suitable for deriving the set of 'preferred' operator actions predicted (expected according to the nominal model) for a given operational context. For the flight deck, such models may be adapted from extant Advanced Qualification Program (AQP) models (U.S. FAA, 1995) and validated in high fidelity simulations. (The original CATS B757 model, however, was initially derived from a training program at a major airline, together with expert input from line pilots.)

CATS identifies two types of errors: errors of omission, and errors of commission. It further identifies errors of commission that result when the 'right action' is performed with the 'wrong value.' CATS does not base these determinations on a 'formulaic' representation of how such errors would appear in a trace of operator activities, nor attempt to further classify errors (e.g., 'reversals') as in some research on formal methods for identifying potential errors (Wright, Fields, and Harrison, 1994). Indeed, this would be difficult, given that the CATS model does not represent the 'steps' of procedures explicitly as 'step A follows step B;' instead it represents procedures implicitly by explicitly specifying the conditions under which operators should preferably perform each action. CATS predicts concurrent actions whenever the current context satisfies conditions for performing two or more activities. CATS interprets concurrent actions whenever the granularity of action data identifies them as such.

Like analysis techniques that rely on a 'reflection' of the task specification in a formal model of a system (Degani and Heymann, 2000, Sherry *et al.*, 2001), CATS relies on a correctly functioning system to reflect the results of actions (or inaction) in its state. CATS identifies errors by using information in the CATS model that enables it to assess actions (or the lack thereof, in the case of omissions) in light of the current operational context and the future context formed as a result of operator action (or inaction). Thus, one might view the CATS error detection scheme as 'closing the loop' between a representation of correct task performance and the controlled system, and evaluating feedback from the controlled system to ensure it 'jibes' with correct operator activities. Given that the system is operating normally and providing 'good data,' this is a powerful concept.

Crew Activity Tracking System (CATS)

CATS implements a methodology for activity tracking in a computer-based system that has been validated to work in real time (Callantine, Mitchell, and Palmer, 1999). Figure 1 generically depicts information flow in CATS, between a controlled system and CATS, and between CATS and applications based on it. As described above, CATS uses representations of the current state of the controlled system and constraints imposed by

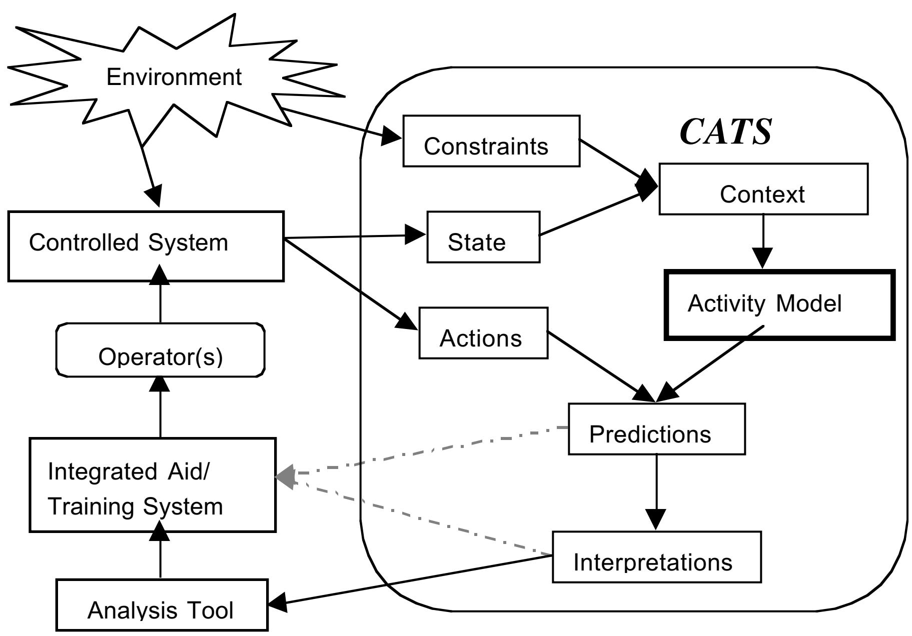

Figure 1. Information flow within and between CATS and a generic human-machine system, and applications to error analysis, aiding, and training.

the environment (including performance limits on the controlled system) to derive the current operational context. CATS then uses this representation to generate predictions from its model of operator activities. CATS compares detected operator actions to its predicted activities, and it assesses actions that it cannot immediately interpret as matching a prediction by periodically referencing the activity model until it receives enough new context information to disambiguate possible interpretations.

Thus, two threads comprise the activity tracking methodology as implemented in CATS: a 'prediction thread' responsible for generating the context information necessary to predict nominal activities, and an 'interpretation thread' that interprets operator actions. Displays of the resulting interpretations—together with displays for visualizing the input data, current operational context, and activity model—comprise a CATS-based analysis tool (Callantine, 2000a, 200b). Predictions and interpretations supply the information necessary for an aid that is integrated into the displays of the controlled system (Callantine, 1999) or, in the case of a tutoring system, a high-fidelity simulation of the controlled system.

CATS Implementation for Flight Data Error Detection

The following subsections specifically describe the implementation of CATS for detecting pilot errors from flight data. The first is devoted to the flight data itself. The second illustrates a portion of the CATS model, and the third describes how CATS generates the current operational context using a representation of ATC clearance constraints. The CATS model fragment is relevant to an example of CATS detecting pilot errors presented in the fourth subsection.

The following subsections all assume some knowledge of commercial aviation and a B757-style autoflight system. The basic scheme is that pilots first program the flight plan into the FMS via the CDU. After engaging the autopilot (or flight director) and the autothrottles, they interact with aircraft's Mode Control Panel (MCP), setting tactical targets and engaging pitch, roll, and thrust modes as required to comply with air traffic control clearances. High-level modes such as Lateral Navigation (LNAV) and Vertical Navigation (VNAV) track the FMS-programmed plan; other modes, such Flight Level Change (FL CH), achieve a tactical target state (the MCP target altitude, in the case of FL CH). A detailed description of the Boeing 757 autoflight system mode usage is provided in Callantine, Mitchell, and Palmer (1999); see Billings (1997), Sarter and Woods (1995), and Wiener (1989) for discussions of mode errors and automation issues.

B757 ARIES Flight Data

The NASA Langley B757 ARIES aircraft, with its onboard Data Acquisition System (DAS), provided the flight data for this research (Figure 2). The DAS collects data at rates in excess of 5 Hz, using onboard computers that perform sensor data fusion and integrity checking. In future applications such functionality may be required within CATS. Table 1 shows the collection of values that comprise the data set. The data include information from important cockpit systems. The rightmost column of Table 1 shows data CATS derives from the sampled values using filtering techniques. Included are crew action events CATS derives from the values of control states. Target value settings on the MCP are derived with 'begin' and 'end' values, as in formal action specification schemes (Wright, Fields, and Harrison, 1996). Like the initial CATS research (Callantine and Mitchell, 1994), this application focuses on interactions with the autoflight system MCP, so it only uses some of the available data.

Absent from data in Table 1 are important flight management system (FMS) data, including actions pilots perform using the flight management computer (FMC) control and display units (CDUs). This is a shortcoming of the B757 ARIES DAS that future research seeks to rectify. In the interim, tracking CDU interactions with CATS is feasible with the NASA Ames Advanced Concepts Flight Simulator (ACFS), a full-motion, high-fidelity glass cockpit simulator (Callantine, 2000), and its desktop

Figure 2. Data Acquisition System (DAS) onboard the NASA B757 ARIES aircraft (inset).

counterpart, the 'miniACFS.' To detect entries that a pilot types into the CDU scratchpad, CATS uses a parsing mechanism. It detects CDU keystrokes and 'releases' a fully-formed action (e.g., 'crossing restriction entered') when the character string created exactly matches a value that CATS expects, or when it can be determined not to match any related value. Thus, unlike CATS in general, this parsing process *does* incorporate a rudimentary *a priori* model of what sorts of errors a pilot might make. This is an area of further research. Also absent from Table 1 are data concerning ATC clearances. For the present application, cockpit observations provide required clearance information.

CATS Model of B757 Navigation Activities

Figure 3 depicts a fragment of the CATS model used to detect errors from B757 ARIES data. The model decomposes the highest level activity, 'fly glass cockpit aircraft,' into sub-activities as necessary down to the level of pilot actions. Figure 3 illustrates eight actions. All actions derivable from the data are included in the full model. Each activity in the model is represented with conditions that express the context under which the activity is nominally preferred, given policies and procedures governing operation of the controlled system. The parenthesized numbers in Figure 3 refer to Table 2, which lists the 'and-or trees' that comprise these rules.

For comparison to other work that considers human errors involved with CDU manipulations (e.g., Fields, Harrison, and Wright, 1997), the model fragment in Figure 3 shows just one of numerous FMS configuration tasks. Note that a CATS model can also include cognitive, verbal, and perceptual 'activities,' but CATS can only predict, not interpret, activities for which no confirmatory data exists. Thus, such activities are not relevant to the present application.

Table 1. B757 ARIES data used in the present research, including derived states and action events (rightmost column). Some variables appear multiple times, because the B757 ARIES DAS collects them from multiple sources.

Time variables
time
time1
time2
time3

Environmental information
total_air_temp
true_wind_dir
wind_speed

AC position/attitude
baro_alt
baro_corr
flight_path_angle
ground_speed
computed_airspeed
calibrated_airspeed
mach
magnetic_heading
magnetic_track_angle
pitch_angle
radio_altitude
roll_angle
true_track_angle
iru_potential_vert_speed
hybrid_lat
hybrid_lon

AC configuration/controls
left_engine_epr
right_engine_epr
flap_pos
speed_brake_handle
left_throttle_pos
right_throttle_pos
gross_weight

MCP target values
sel_mcp_altitude
sel_mcp_heading
sel_mcp_speed
sel_mcp_vert_speed
mcp_flare_retard_rate
sel_mcp_mach

MCP bank angle settings
bank_angle_lim_flaps_25
bank_angle_lim_flaps_15
bank_angle_lim_auto

NAV/COMM data
dme_range
left_dme_freq
right_dme_freq
left_dme_dist
right_dme_dist
left_vhf_freq
right_vhf_freq

FMC data
fmc_target_airspeed
fmc_selected_altitude
fmc_selected_airspeed
fmc_selected_mach
fmc_crz_altitude
fmc_eta
fmc_desired_track
fmc_wpt_bearing
fmc_cross_track_dist
fmc_vert_dev
fmc_range_to_alt
fmc_wide_vert_dev

AFDS states
ap_cmd_ctr_engd
ap_cmd_cen_gc_huh
ap_cmd_cen_gr_huh
left_ap_cmd_engd
ap_cmd_left_engd
right_ap_cmd_engd
ap_cmd_right_engd
ap_cmd_center_engd
ap_cws_center_engd
ap_cws_left_engd
ap_cws_right_engd
ap_in_control
fd_c_on
fd_fo_on
fd_on_c
fd_on_fo

AFDS switches
ap_cmd_center_reqd
ap_cmd_right_reqd
ap_cws_center_reqd
ap_cws_left_reqd
ap_cws_right_reqd
ap_cmd_left_reqd

AFDS modes
fl_ch_engd
hdg_hold_engd
hdg_sel_engd
land_2_green
land_3_green
alt_hold_engd
vnav_armed_engd
lnav_armed_engd
speed_mode_engd
thrust_mode_engd
loc_engd
vert_spd_engd
apprch_armed_engd
loc_armed_engd
back_course_armed_engd
glideslope_engd

MCP Speed display status
mcp_speed_display_blank

Autothrottle
at_armed

MCP switches
hdg_sel_reqd
hdg_hold_reqd
lnav_reqd
vnav_reqd
spd_reqd
apprch_reqd
loc_reqd
alt_hold_reqd
vs_mode_reqd
fl_ch_reqd
thrust_mod_reqd

IAS/Mach toggle
mach_toggled

Crew Alert levels
crew_alert_level_a
crew_alert_level_b
crew_alert_level_c

Status data
eec_valid
engine_not_out

FMC-A/T internal data
fmc_at_mach_mode_reqd
fmc_at_airspeed_mode_reqd
fmc_active_climb
fmc_climb_mode_reqd
fmc_active_cruise
fmc_con_mode_reqd
fmc_crz_mode_reqd
fmc_active_descent
fmc_display_annunc_on
fmc_eng_ident_1
fmc_eng_ident_2
fmc_eng_ident_3
fmc_eng_ident_4
fmc_eng_ident_5
fmc_eng_ident_6
fmc_eng_ident_7
fmc_eng_ident_8
fmc_eng_ident_9
fmc_eng_ident_10
fmc_ga_mode_reqd
fmc_idle_thr_reqd
fmc_msg_annunciated
throttle_retard_reqd
pitch_speed_control_engd
vnav_operational
lnav_operational
tmc_valid

VNAV submodes
fmc_vnav_speed_operational
fmc_vnav_path_operational
fmc_vnav_alt_operational

Thrust ratings
fmc_rating_1_reqd
fmc_rating_2_reqd
fmc_offset_annunciated
fmc_throttle_dormant_reqd
fmc_thr_mode_reqd
fmc_to_mode_reqd
req_1_valid_resv
req_2_valid_resv

Derived states
vert_speed
alt_cap_engaged
spd_win_auto_chng
ap_cmd_engd

Derived MCP actions
set MCP hdg
set MCP alt
set MCP spd
set MCP mach
set MCP vs
hdg sel press
hdg hold press
lnav press
vnav press
spd press
apprch press
loc press
alt hold press
vs mode press
fl ch press
thrust mode press
mach toggled
c ap cmd switch press
l ap cmd switch press
r ap cmd switch press
arm autothrottles

Other derived actions
tune left VHF
tune right VHF
set flaps
set spoilers

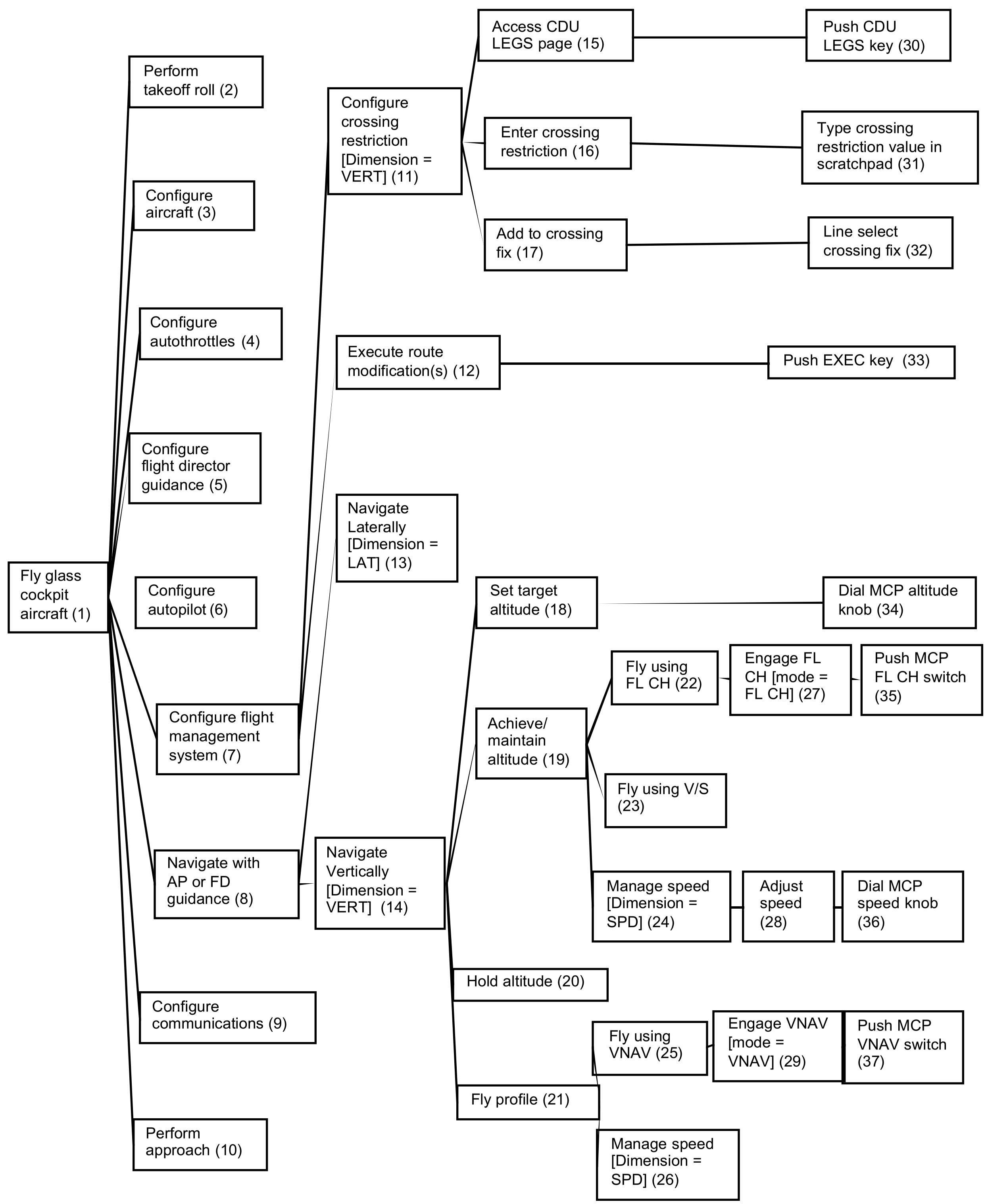

Figure 3. Fragment of CATS model for B757 operations.

Table 2. 'And-or' trees of conditions under which the CATS model in Figure 3 represents activities as 'nominally preferred.' CATS predicts an activity when its conditions, plus all the conditions of its parent activities, are satisfied by the current operational context.

(1) start-of-run
(2) (not above-runway-elevation)
(3) (and (not above-clean-speed) (not flight-surfaces-within-limits) (not gear-within-limits))
(4) (not autothrottle-armed)
(5) (not flight-director-on)
(6) [(and (not autopilot-cmd-mode-engaged) above-1000-feet-AGL)]
(7) (or (not programmed-route-within-limits) route-uplink-received)
(8) (and above-1000-feet-AGL (or autopilot-cmd-mode-engaged flight-director-on))
(9) (not comm-frequency-within-limits)
(10) (or approaching-glideslope-intercept-point approach-localizer-intercept-point)
(11) (not crossing-restriction-within-limits)
(12) route-modifications-within-limits
(13) (or autopilot-cmd-mode-engaged flight-director-on)
(14) (or autopilot-cmd-mode-engaged flight-director-on)
(15) (not cdu-page-LEGS)
(16) (and cdu-page-LEGS (not crossing-restriction-built))
(17) (and cdu-page-LEGS crossing-restriction-built)
(18) (not mcp-altitude-within-limits)
(19) (or (and (not current-altitude-within-limits) (not profile-within-limits-for-now)) expedite-needed)
(20) (and current-altitude-within-limits (not profile-within-limits-for-now))
(21) profile-within-limits-for-now
(22) (or (not altitude-close-to-target) expedite-needed)
(23) altitude-close-to-target
(24) (or fl-ch-engaged vs-engaged)
(25) profile-within-limits-for-now
(26) vnav-engaged
(27) (not fl-ch-engaged)
(28) (not target-speed-within-limits)
(29) (and (not vnav-engaged) (not capturing-required-altitude))
(30) (not cdu-page-LEGS)
(31) (not crossing-restriction-built)
(32) crossing-restriction-built
(33) route-modifications-within-limits
(34) (not mcp-altitude-within-limits)
(35) mcp-altitude-within-limits
(36) (not target-speed-within-limits)
(37) mcp-altitude-within-limits

Several important features of the CATS model deserve mention. First, when using the model to predict the currently preferred set of activities, CATS searches the model top-down, so that higher level activities 'subsume' their children (i.e., the conditions on an activity must be met before CATS can predict any of its children). Thus, CATS makes predictions and interpretations at every level of abstraction the model represents. Second, the model itself is 'memoryless.' Given some context, CATS can predict what the operators need to do (as discussed below, however, historic information *is* contained in the context, and in some cases does impact the CATS predictions). Third, the model can be structured to represent the activities of an individual operator, or a team of operators (cf. Fields, Harrison, and Wright, 1997); either way, CATS is capable of detecting errors that relate to assigned operator roles and responsibilities. Fourth, the model contains information—beyond that provided by its structure and preference conditions—to support error detection. One type of information concerns the automation mode that should be operational if a mode-engagement action was successfully invoked. Another concerns the 'dimension' of the operational context that an activity addresses.

Representation of ATC Clearance Constraints for Context Generation

Environmental constraints play a key role in defining the goals that shape worker behavior in complex sociotechnical systems (Vicente, 1999). CATS also relies on a representation of environmental constraints to construct a representation of the current operational context (Figure 1). These factors motivated recent research on a symbolic representation of the constraints ATC clearances impose on flight operations (Callantine, 2002). Figure 4 shows the representation, which represents three key dimensions of constraints: vertical, lateral, and speed. CATS employs a rule base that enables it modify this constraint representation to reflect the constraints imposed (or removed) by each new ATC clearance.

As discussed in Callantine (2002), CATS defines context from a human operator's perspective to be the *situation* plus any activities the operator is engaged in performing. The situation is defined as the system's *state*, together with environmental constraints and all salient relationships between the state and constraints. Each of these elements is additionally considered to have historic, current, and planned (or predicted) future components. States and constraints are also decomposed hierarchically at multiple levels of abstraction as necessary.

CATS uses a representation of context of this form to generate a summary of the current operational context suitable for evaluating the conditions under which activities are preferred, in order to predict activities, and for determining whether an operator action it did not expect is in error. Whenever the state or constraints change, CATS examines the salient relationships to generate a set of 'context specifiers' that summarizes the current operational context; these are the descriptive clauses that appear in the conditions listed in Table 2. CATS also uses the symbolic constraint representation to maintain a record of compliance with constraints. This is important not only for context generation, but also for logging flight path deviations.

Error Detection Example

The report now presents an example of CATS detecting errors from B757 ARIES flight data collected during recent flight test activities. Although the data are real, in the flight test environment, strict procedures about how the pilots should preferably fly the airplane are unreasonable. Nonetheless, by imposing the model depicted in part in Figure 3, CATS was able to detect errors, and the errors were not contrived. While the errors CATS detects are insignificant, because they in no way compromised safety, the exercise nonetheless demonstrates the viability of CATS for error detection. It should be noted that, in this application, as the following 'snapshots' show, CATS runs at between twelve and twenty-two times real time.

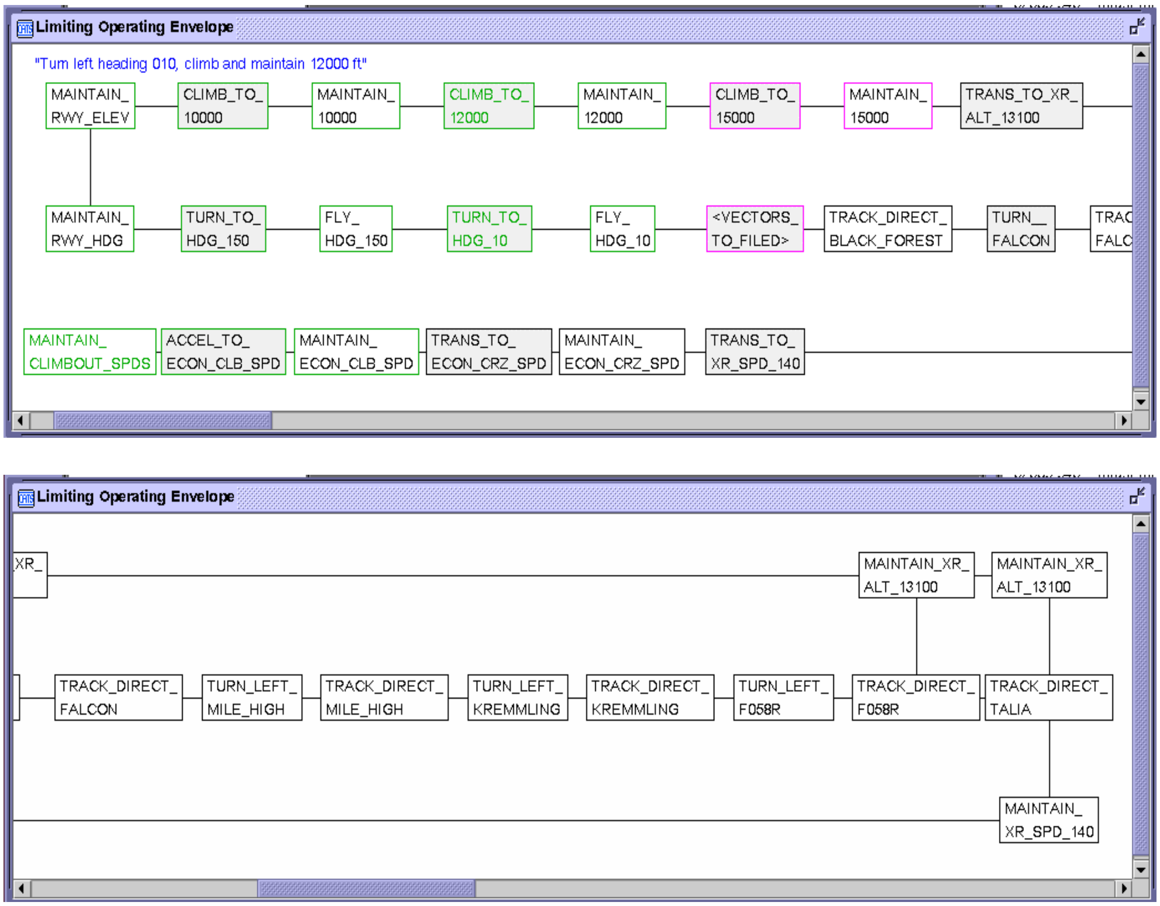

Figure 4. Snapshot of a CATS representation of environmental constraints constructed from the filed flight plan, and modified according to constraints conveyed by ATC clearances.

Figure 5 shows the CATS interface at the start of the scenario (Scenario Frame 1). The crew has just received a clearance to “climb and maintain 16,000 feet.” CATS modifies its representation of ATC clearance constraints accordingly, and using the updated context, predicts that the crew should set the new target altitude on the MCP by dialing the MCP altitude knob.

In Scenario Frame 2 (Figure 6), a pilot instead pushes the VNAV switch. Because CATS has not predicted this action, it cannot interpret the action initially. CATS instead continues processing data.

In Scenario Frame 3 (Figure 7), CATS has received enough new data to interpret the VNAV switch press action. Had the action been correct, the autoflight system state would have reflected this by engaging the VNAV mode and commencing the climb. However, VNAV will not engage until a new target altitude is set. To assess the VNAV switch press with regard to the current context, in which airplane is still in ALT HOLD mode at 12,000 feet, CATS searches its model to determine if any parent activities of the VNAV switch press contain information linking the action to a specific context. CATS finds that the ‘engage VNAV’ activity should reflect VNAV mode engagement in the current context (see Figure 3). Because this is not the case, CATS flags the VNAV switch press as an error. Meanwhile, CATS still expects the crew to dial the MCP altitude knob.

In Scenario Frame 4 (Figure 8), a pilot does begin setting the MCP altitude. CATS interprets this action as matching a current prediction, but with an incorrect value, as the altitude setting has not yet reached 16,000.

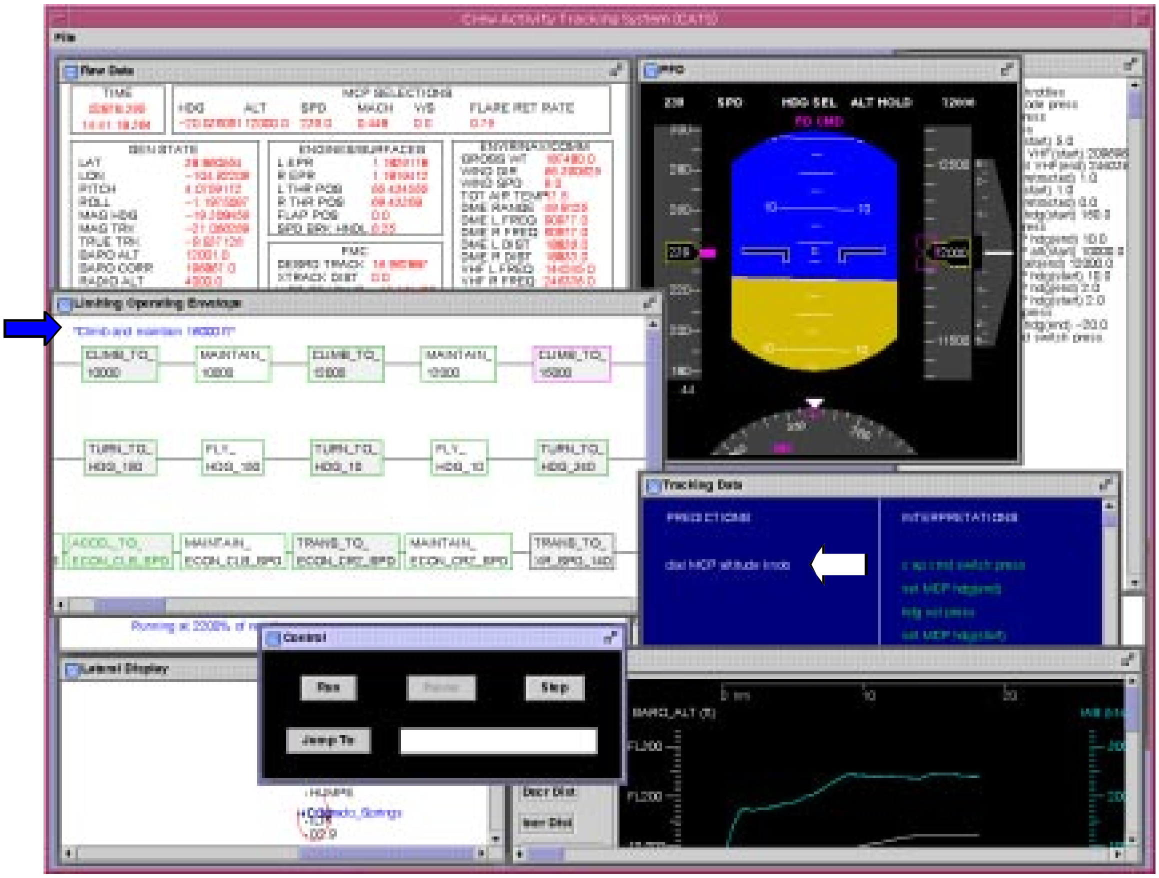

Figure 5 (Scenario Frame 1). In response to a clearance to climb, CATS predicts the crew should set the new target altitude on the MCP by dialing the MCP altitude knob._

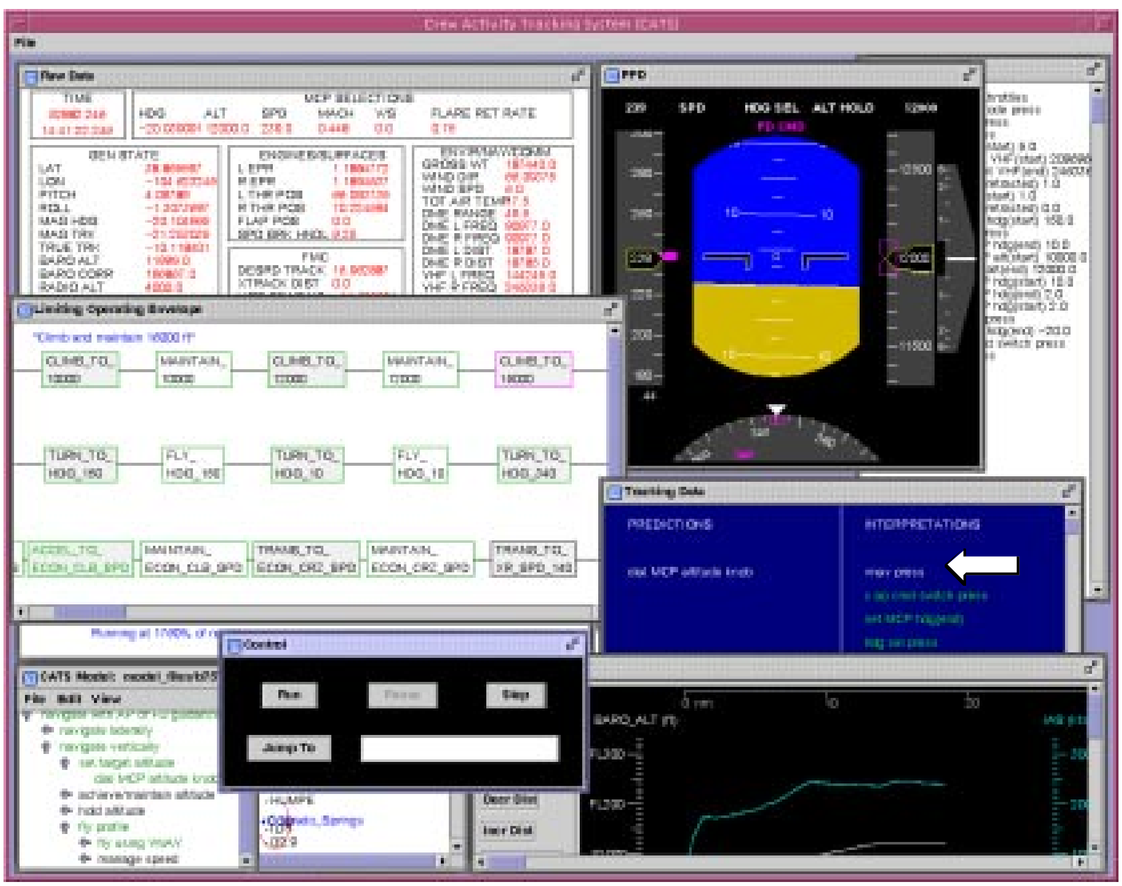

Figure 6 (Scenario Frame 2). CATS detects that a crew member pressed the VNAV switch instead.

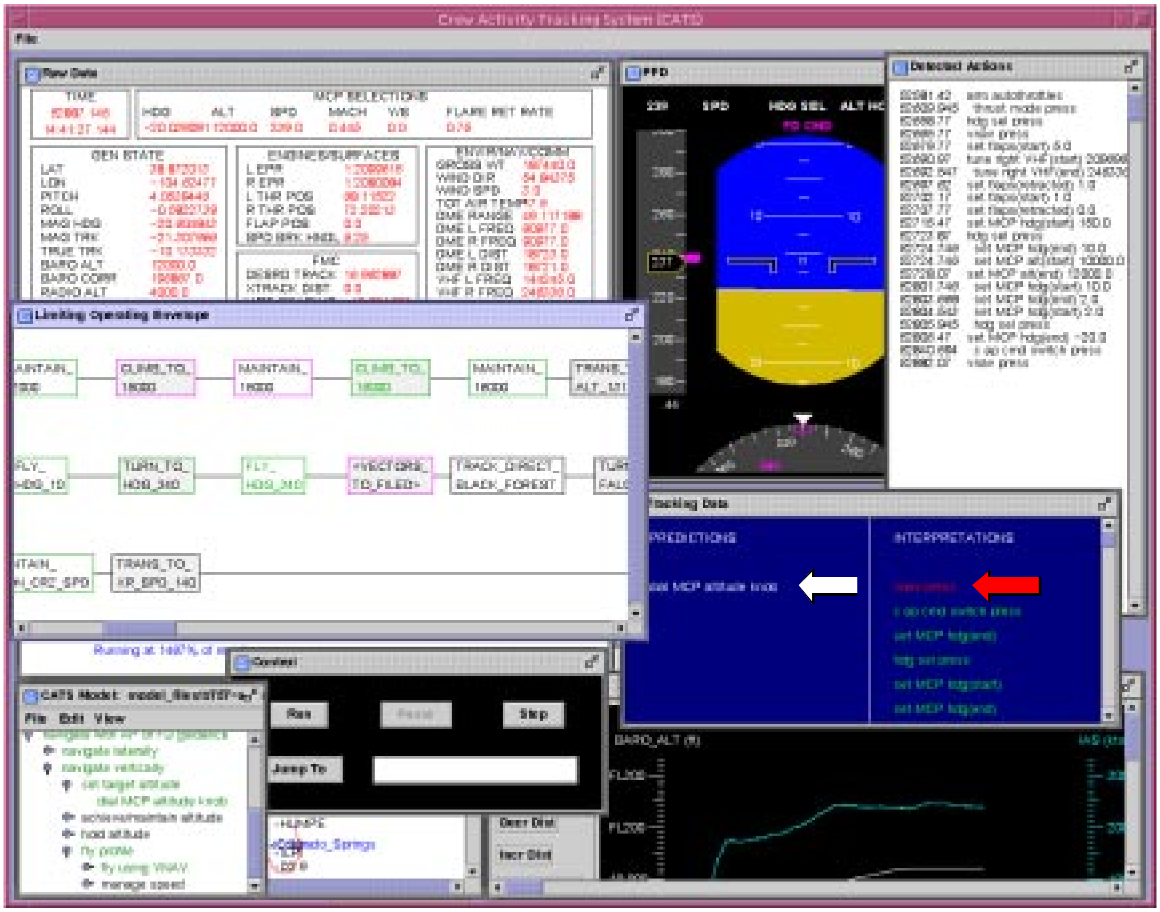

Figure 7 (Scenario Frame 3). CATS cannot reconcile the VNAV switch press with the current context, and therefore flags it as an error. CATS is still expecting the crew to dial the MCP altitude knob.

CATS does not flag this action as a ‘wrong value’ error, however, because it is only the start of the altitude setting. CATS continues to predict ‘dial MCP altitude knob’ because the context specifier ‘mcp-altitude-within-limits’ is not generated when the current MCP target altitude is compared to the value specified by the representation of ATC constraints (see Figure 3 and Table 2).

In Scenario Frame 5 (Figure 9), one pilot pushes the VNAV switch a second time before the altitude setting is complete. As the other pilot completes the altitude setting, CATS interprets the end of the altitude setting action as matching its prediction.

In Scenario Frame 6 (Figure 10), CATS detects that a pilot has pressed the FL CH switch (perhaps to begin the climb in FL CH mode, since VNAV did not engage). Because the MCP target altitude is now properly set, CATS predicts the crew should engage VNAV, which is preferred according to the CATS model.

CATS detects a second FL CH switch press in Scenario Frame 7 (Figure 11). Perhaps a pilot performed this action as ‘insurance’ to engage a mode to begin the climb. Because FL CH mode engages, and this is reflected in CATS’ representation of the current context, CATS interprets both FL CH switch presses as correct acceptable alternative actions. By this time, CATS has also flagged the second VNAV switch press as an error.

In the final frame of the scenario (Scenario Frame 8, Figure 12), the aircraft has begun climbing in FL CH mode. At this point the crew opts to engage VNAV mode. At last, CATS detects the predicted VNAV switch press and interprets it as correct.

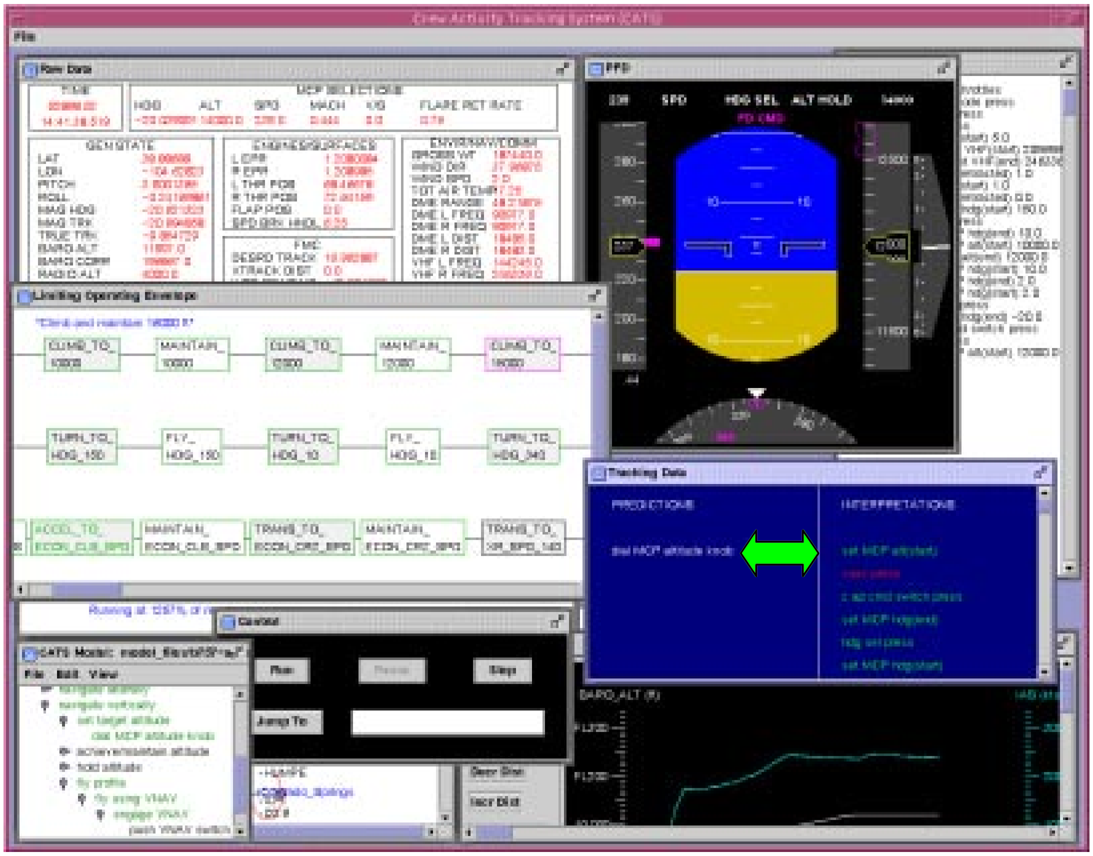

Figure 8 (Scenario Frame 4). CATS detects a pilot starting to dial the MCP altitude, and interprets it as matching its prediction, but with the wrong value (This is not an error, because the action is only the start of the altitude setting).

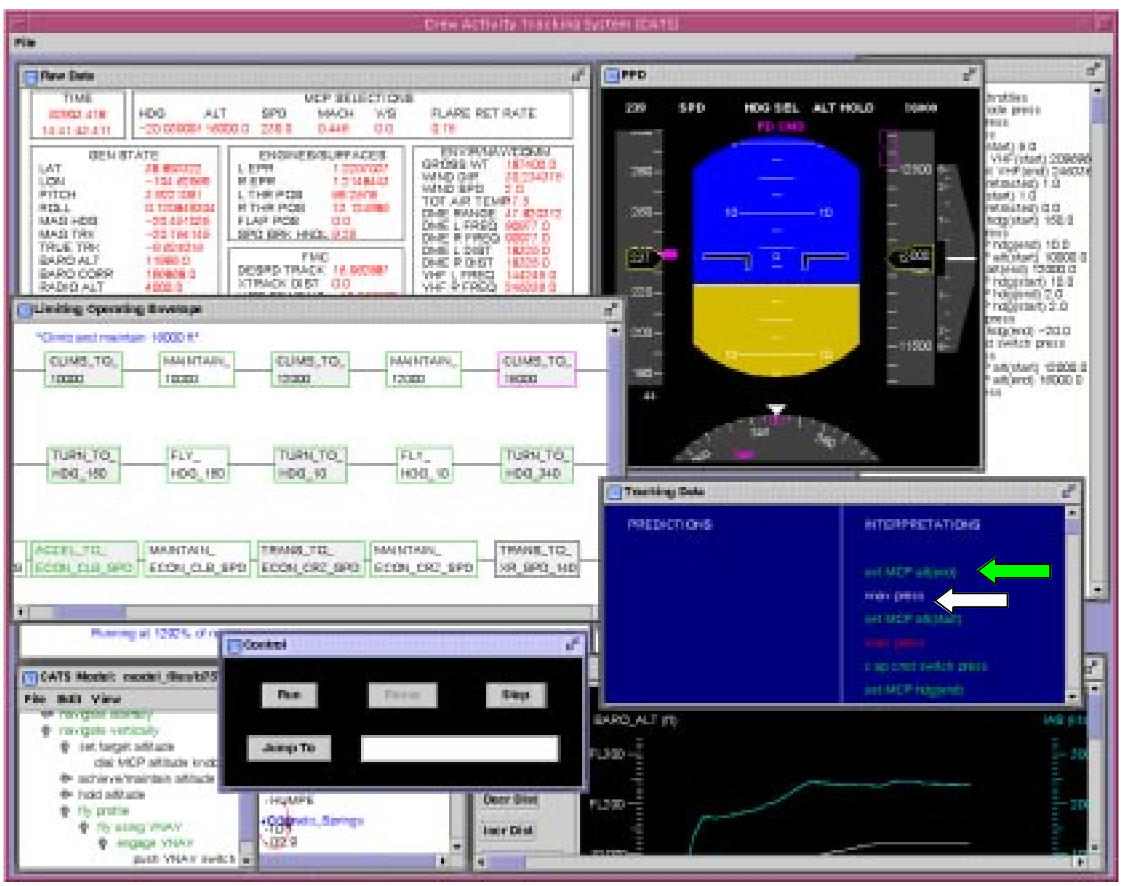

Figure 9 (Scenario Frame 5). CATS detects a second VNAV switch press, prior to the time when the altitude setting is finished.

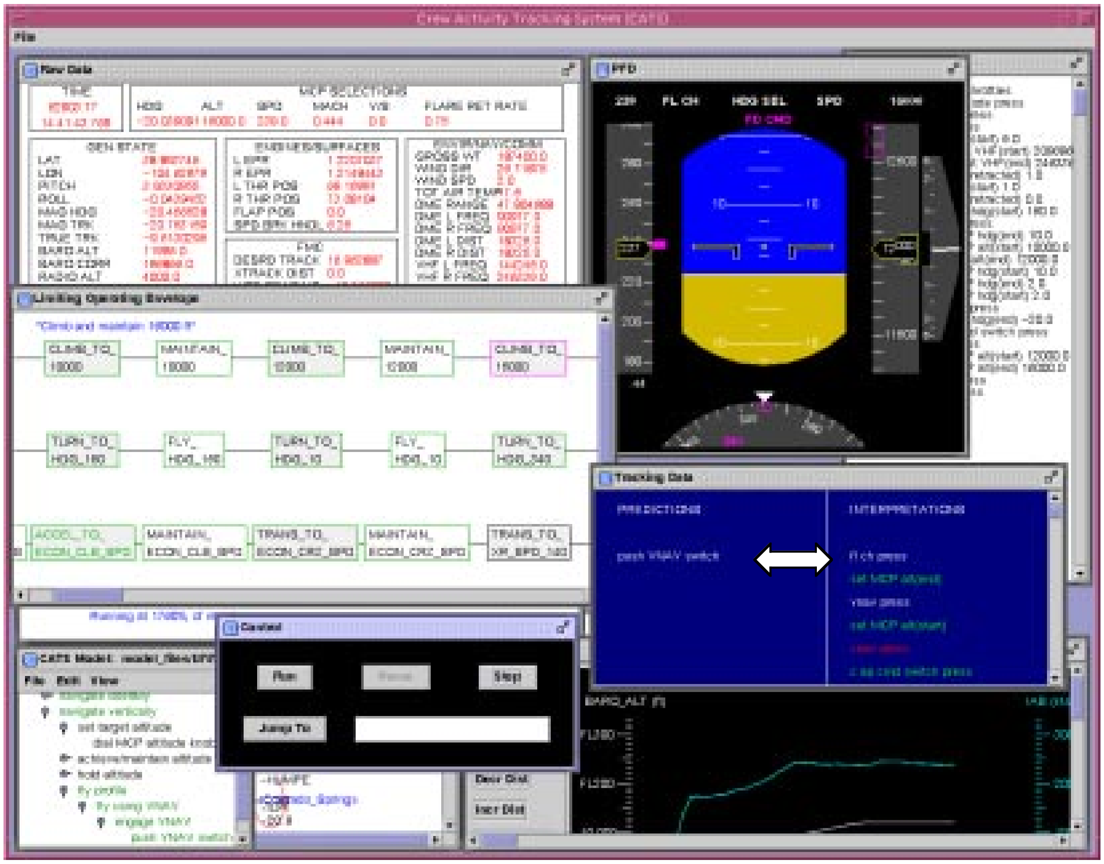

Figure 10 (Scenario Frame 6). CATS detects that the crew has now opted to engage FL CH mode by pressing the FL CH switch; but because the altitude is now properly set, CATS now predicts the crew should push the VNAV switch to engage VNAV (the preferred mode according to the CATS model).

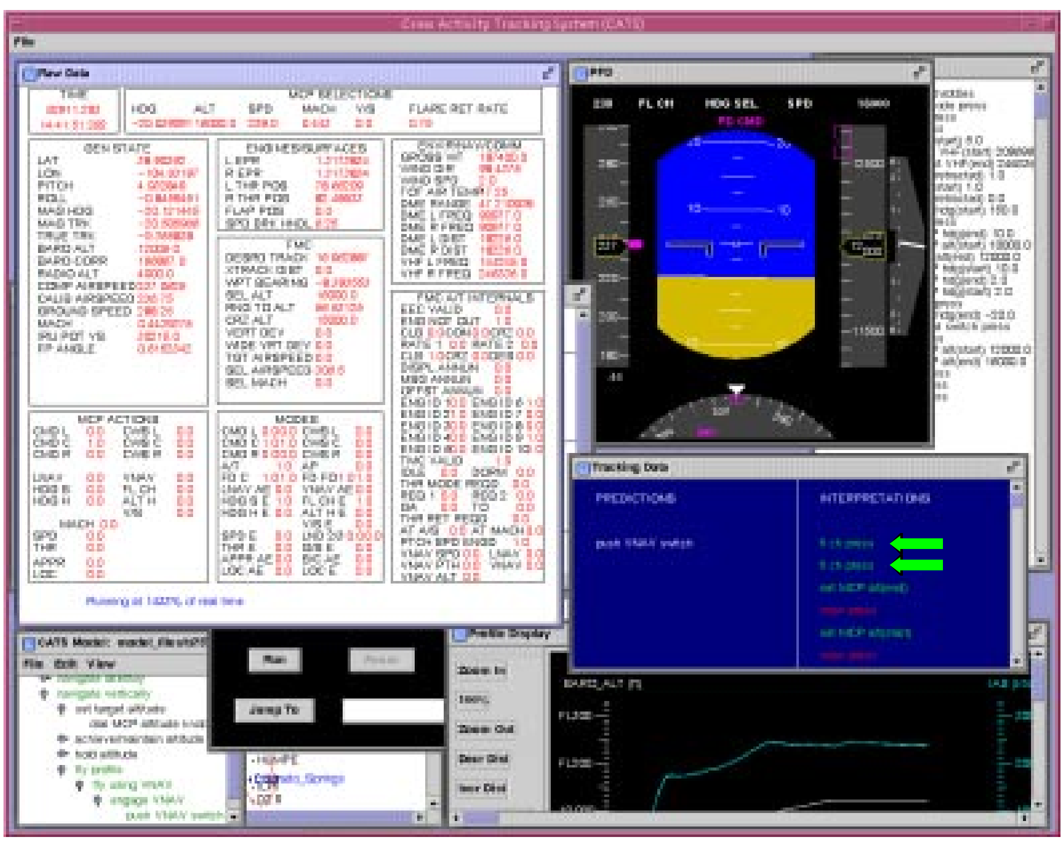

Figure 11 (Scenario Frame 7). CATS detects a second ‘insurance’ FL CH switch press, and interprets it as acceptable as it did the first FL CH switch press.

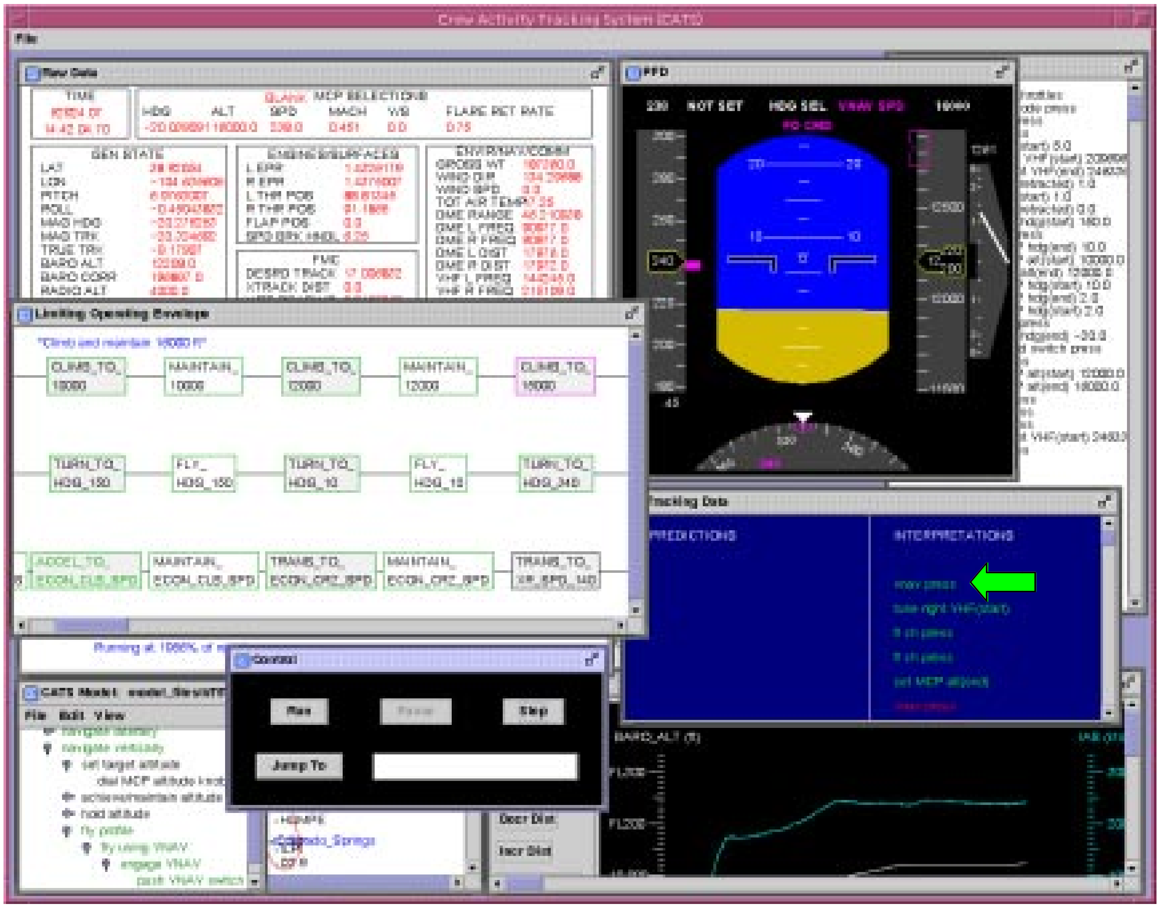

Figure 12 (Scenario Frame 8). The crew opts to engage VNAV; CATS detects the predicted VNAV switch press and interprets it as correct (elapsed time from Scenario Frame 1 is ~42 secs).

Conclusions and Future Research

The above example demonstrates that CATS can detect errors from flight data. Although the errors CATS detects are inconsequential, this research indicates CATS can provide contextual information useful for disambiguating the causes of deviations or unusual control actions that arise in incident or accidents. Discoveries made using CATS can be incorporated into training curricula by connecting a CATS-based training system to a simulator and allowing pilots to ‘fly’ under conditions that correspond the actual context of an error-related event. Such capabilities are also useful outside the airline arena as they support both fine-grained cognitive engineering analyses and human performance modeling research.

Using CATS with flight data collected at ‘continuous’ rates results in better performance. Event-based data, such as those available from the NASA ACFS, require more complicated interpolation methods to avoid temporal ‘gaps’ in the CATS representation of context that can adversely affect CATS performance. Important directions for further research involve improving the coverage of flight data to include the FMS and CDUs, as well as work on methods to automatically acquire ATC clearance information. This research indicates that, if CATS has access to data with full, high-fidelity coverage of the controlled system displays and controls, it can expose the contextual nuances that surround errors in considerable detail.

References

Babbage, C. (1961). *On the principles of the calculator and other seminal writings by Charles Babbage* [b.1792-d.1871] *and others*. In P. Morrison & E. Morrison (Eds.), New York, NY: Dover.

Billings, C. (1997). *Human-centered aviation automation*. Mahwah, NJ: Lawrence Erlbaum Associates.

Busse, D., and Johnson, C. (1998). Modeling human error within a cognitive theoretical framework. In F. Ritter & R. Young (Eds.), *Proceedings of the 2nd European Conference of Cognitive Modeling (ECCM-98),* Nottingham, UK: Nottingham University Press, 90-97.

Byrne, M., and Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science, 21*(1), 31-61.

Callantine, T. (1999). An intelligent aid for flight deck procedure performance: The Crew Activity Tracking System "Task Caddy." *Proceedings of the 1999 IEEE Conference on Systems, Man, and Cybernetics,* Tokyo, October, 1095-1100.

Callantine, T. (2000a). *A glass cockpit crew activity analysis tool*. SAE Technical Paper 200-01-5522. Warrendale, PA: SAE International.

Callantine, T. (2000b). Online analysis of high fidelity simulation data. *Proceedings of the International Conference on Cognitive Systems Engineering for Process Control 2000,* Taejon, Korea, November, 175-182.

Callantine, T. (2001a). Agents for analysis and design of complex systems. *Proceedings of the 2001 International Conference on Systems, Man, and Cybernetics*, Tuscon, AZ, October, 567-573.

Callantine, T. (2001b). The crew activity tracking system: Leveraging flight data for aiding, training, and analysis. *Proceedings of the 20th Digital Avionics Systems Conference*, Daytona Beach, FL, October, 5.C.3-1—5.C.3-12 (CD-ROM).

Callantine, T. (2002). A representation of air traffic control clearance constraints for intelligent agents, *Proceedings of the 2002 IEEE Conference on Systems, Man, and Cybernetics*, Hammamet, Tunisia, October.

Callantine, T., and Mitchell, C. (1994). A methodology and architecture for understanding how operators select and use modes of automation in complex systems. *Proceedings of the 1994 IEEE Conference on Systems, Man, and Cybernetics*, San Antonio, TX, 1751-1756.

Callantine, T., Mitchell, C., and Palmer, E. (1999). *GT-CATS: Tracking operator activities in complex systems*. NASA Technical Memorandum 208788, Moffett Field, CA, NASA Ames Research Center.

Degani, A. and Heymann, M. (2000). *Some formal aspects of human automation interaction*. NASA Technical Memorandum 209600, Moffett Field, CA: NASA Ames Research Center.

Fields, R., Harrison, M., and Wright, P. (1997). *THEA: Human error analysis for requirements definition*. Technical Report 2941997, York, UK: University of York Computer Science Department.

Fields, R., Wright, P., and Harrison, M. (1996). Temporal aspects of usability: Time, tasks and errors. *SIGCHI Bulletin 28*(2).

Johnson, C. (1998). Representing the impact of time on human error and systems failure. *Interacting with Computers, 11,* 53-86.

Johnson, C. (2000). Novel computational techniques for incident reporting. In D. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the 2000 Workshop* (Technical Report WS-00-03), Menlo Park, CA: AAAI Press.

Johnson, C. (2000). Supporting the analysis of human error in national and international incident reporting schemes. In P. Cacciabue (Ed.), *Proceedings of the 19th European Annual Conference on Human Decision Making and Manual Control*, Ispra, Italy, June, 137-144.

Johnson, C. (2001). The London Ambulance Service, computer aided dispatch system: A case study in the integration of accident reports and the constructive design of safety-critical computer systems. *Reliability Engineering and Systems Safety, 71*(3), 311-326.

McGreevy, M. (2001). *Searching the ASRS database using QUORUM keyword search, phrase search, phrase generation, and phrase discovery*, NASA Technical Memorandum 210913, Moffett Field, CA: NASA Ames Research Center.

Mitchell, C. (2000). Horizons in pilot training: Desktop tutoring systems. In N. Sarter & R. Amalberti (Eds.), *Cognitive Engineering in the Aviation Domain*. Mahwah, NJ: Lawrence Erlbaum Associates, 211-251.

Perrow, C. (1984). *Normal accidents: Living with high-risk technologies*. New York, NY: Basic Books.

Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.

Sarter, N., and Woods, D. (1995). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. *Human Factors, 31*(1), 5-19.

Sherry, L, Feary, M., Polson, P., Mumaw, R., and Palmer, E. (2001). *A cognitive engineering analysis of the vertical navigation (VNAV) function*. NASA Technical Memorandum 210915, Moffett Field, CA: NASA Ames Research Center.

Smith, N., Brown, J., Polson, P., and Moses, J. (2001). An assessment of flight crew experiences with FANS-1 controller-pilot data link communication in the South Pacific. *Proceedings of the 4th USA/Europe Air Traffic Management R&D Seminar*, Santa Fe, NM, December.

U.S. Federal Aviation Administration (1995). *Advanced qualification program*, Advisory Circular AC 120-54.

U.S. General Accounting Office (1998). Aviation safety: U.S. efforts to implement Flight Operational Quality Assurance programs, *Flight Safety Digest, 17*(7-9), 1-36.

Vicente, K. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. Mahwah, NJ: Lawrence Erlbaum Associates.

Wiener, E. (1989). *The human factors of advanced technology ("glass cockpit") transport aircraft*. NASA Contractor Report 177528, Moffett Field, CA: NASA Ames Research Center.

Wiener, E., and Curry, R. (1980). Flight deck automation: Promises and problems. *Ergonomics, 23,* 995-1011.

Wright, P., Fields, R., and Harrison, M. (1994). Deriving human-error tolerance requirements from tasks. *Proceedings of the 1st IEEE International Conference on Requirements Engineering*, Colorado Springs, CO, April.

Report Documentation Page	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2002	Technical Memorandum

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Activity Tracking for Pilot Error Detection from Flight Data	728-20-10
6. AUTHOR(S) Todd J. Callantine	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Ames Research Center Moffett Field, California 94035-1000	8. PERFORMING ORGANIATION REPORT NUMBER IH-030
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration	10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA/CR—2002–211406

11. SUPPLEMENTARY NOTES

Point of Contact: Everett Palmer, M/S 262-4, Ames Research Center, Moffett Field, CA 94035
(650) 604-6073

12A. DISTRIBUTION/AVAILABILITY STATEMENT	12B. DISTRIBUTION CODE
Subject Category: 03-01, 63-02 Distribution: Public Availability: NASA CASI (301) 621-0390	

13. ABSTRACT *(Maximum 200 words)*

This report presents an application of activity tracking for pilot error detection from flight data, and describes issues surrounding such an application. It first describes the Crew Activity Tracking System (CATS), in-flight data collected from the NASA Langley Boeing 757 Airborne Research Integrated Experiment System aircraft, and a model of B757 flight crew activities. It then presents an example of CATS detecting actual in-flight crew errors.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Human error detection, Activity tracking, Glass cockpit aircraft	18
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited